ISSN 2085-4579

Analisis Data *Time Series* Menggunakan LSTM (*Long Short Term Memory*) dan ARIMA (*Autocorrelation Integrated Moving Average*) dalam Bahasa *Python*

Adhitio Satyo Bayangkari Karno[1]
[1] Program Studi Sistem Informasi, Universitas Multimedia Nusantara, Tangerang, Indonesia
adh1t10@yahoo.com

Diterima 7 Agustus 2019
Disetujui 15 Juni 2020

***Abstract*—This study aims to predict time series data by using two methods, the first method commonly used is statistics Autocorrelation Integrated Moving Average (ARIMA model) and the second method which is relatively new, namely machine learning Long Short Term Memory (LSTM). Before the data is processed by both methods, data cleaning and data optimization are carried out. Data optimization is a transformation process to eliminate elements of trends and variations from data. The transformation consists of 7 results of a combination from Log processes, Moving Average (MA), Exponential Weigh Moving Average (EWMA), and Differencing (Diff). The seven processes are each used in the ARIMA and LSTM processes. So that 14 predictions will be obtained (7 from the ARIMA process and 7 from the LSTM process). From the 14 prediction results obtained the smallest RMSE value for ARIMA is 2% and the smallest RMSE value for LSTM is 1%. The results of this study using 7 combinations of transformation processes, can increase the level of accuracy of predictions from ARIMA and LSTM. Where the accuracy of LSTM learning machines by using Telkom's stock data has higher accuracy than ARIMA.**

***Index Terms*—Autocorrelation Integrated Moving Average, Differencing, Exponential Weigh Moving Average, Long Short Term Memory, Machine Learning**

I. PENDAHULUAN

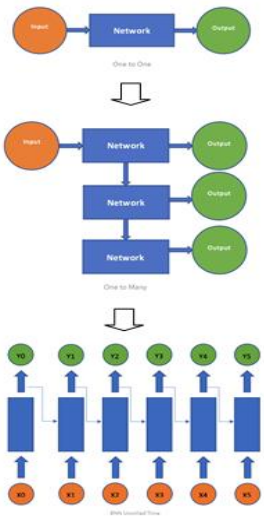
Tidak akan pernah ada yang mengetahui kejadian di masa depan. Namun manusia sebagai mahluk yang paling mulia, dengan akal dan pengetahuannya yang terus berkembang, berupaya untuk dapat memperkirakan kejadian di masa depan dengan tingkat kesalahan sekecil mungkin. Perkiraan ini sangat diperlukan dalam kehidupan manusia agar dapat mempersiapkan semua kemungkinan yang dapat terjadi dengan harapan hidup yang lebih baik dari masa sebelumnya. Untuk dapat memperkirakan masa depan manusia akan belajar dari pengalaman masa lalunya. Pengalaman masa lalu dalam hal ini adalah data runtun waktu (*time series*) yang akan dipergunakan dalam pengolahan data penlitian ini.

Sampai saat ini metode yang masih banyak digunakan dan lebih populer untuk memprediksi data runtun waktu adalah Metoda statistik ARIMA (*Autoregressive Integrated Moving Average*). Dalam ARIMA penentuan model ini ditentukan melalui pencarian nilai parameter dari *Auto-Regressive* (p), *Integrated* (d), *Moving Average* (q) melalui grafik *Autocorrelation Function* (ACF) dan *Partial Autocorrelation Function* (PACF). Penentuan nilai p,d dan q melalui analisa grafik ACF dan PACF masih memberikan tingkat kesulitan untuk memperoleh nilai p, d dan q yang tepat. Sehingga analisa menggunakan ARIMA masih perlu peningkatan akurasi serta menurunkan nilai kesalahan (*error*).

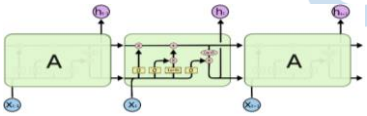
Dalam penelitian ini selain metode statistik ARIMA, akan di pergunakan metoda mesin belajar yang juga mampu mengolah data runtun waktu yaitu *Long Short Term Memory* (LSTM), dengan terlebih dahulu melakukan beberapa proses awal (*cleansing, transformasi, smoothing dan differencing*). Dengan harapan dari kombinasi proses awal pada kedua metoda ini akan menghasilkan tingkat akurasi yang lebih baik.

Berawal dari *neural network* (NN) sederhana yang hanya terdiri dari input, model *network* dan *output* (*one to one*), model ini hanya dapat diterapkan untuk jenis linier regressi dan klasifikasi. Pengembangan selanjutnya adalah model *one to many*, dimana output dari proses sebelumnya akan dipergunakan sebagai input selanjutnya dari proses yang sama. Proses perulangan ini berlangsung dengan input yang tidak berubah dan model *network* yang tetap dari satu iterasi ke iterasi selanjutnya. Karena pengolahan data bersifat runtun waktu, maka diperlukan adanya keterlibatan dengan data sebelumnya, dan *output* yang dihasilkan akan dipergunakan juga sebagai input proses selanjutnya.

Model kemudian dikembangkan lagi menjadi *Recurrent Neural Network* (RNN), dalam model ini setiap data input yang diproses akan melibatkan data sebelumnya [3]. Dalam desain RNN terdapat *multi layer neural network* dengan melibatkan perulangan dalam pemrosesan data yang pada umumnya adalah data runtun waktu. Keterlibatan *multi layer* inilah maka RNN digolongkan sebagai *deep learning*. Berbeda dengan *neural network* kovensional, desain *multi layer* RNN memiliki kemampuan menyimpan memori yang dapat dipergunakan untuk proses selanjutnya. Hal ini sangat diperlukan dalam pemrosesan data runtun waktu yang memiliki keterikatan dengan data sebelum dan sesudahnya.



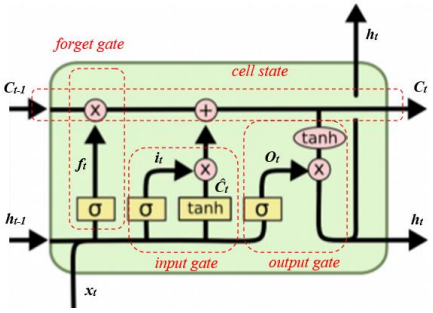
Gambar 1. Perkembangan NN menuju RNN [3]



Gambar 2. Gambar desain jaringan sel LSTM [2]

Namun, RNN dengan runtunan data yang panjang akan menimbulkan situasi dimana nilai gradien yang dipergunakan untuk memperbaharui bobot baru akan bernilai nol (*vanishing gradient*). Hal ini menyebabkan RNN memiliki memori yang hanya mampu mengelola data dengan ketergantungan yang pendek [4].

Untuk mengatasi masalah itu maka dikembangkan sel LSTM. Sel LSTM terdiri dari kombinasi beberapa gerbang (*gate*) yang lebih rumit agar memiliki jaringan blok memori yang mampu mengelola data dengan ketergantungan yang panjang [2].



Gambar 3. Desain sel LSTM

Terdapat 4 kombinasi gerbang dalam satu sel LSTM yaitu:

- Gerbang yang berfungsi memutuskan apakah masukan x_t dan keluaan h_{t-1} akan diteruskan ke *cell state* atau tidak (*forget gate*).

$$f_t = \sigma\,(W_f\,.[h_{t-1}\,, x_t\,] + b_f\,)$$

- Gerbang masukan (*input gate*) dengan dua fungsi aktivasi (sigmoid dan tanh), menentukan bagian mana yang akan diperbaharui.

$$i_t = \sigma\,(W_i\,.[h_{t-1}\,, x_t\,] + b_i\,)$$
$$\tilde{C}_t = \tanh\,(W_C\,.[h_{t-1}\,, x_t\,] + b_C\,)$$

- Gerbang *cell state* untuk mengupdate masukan C_{t-1} (nilai lama) dengan nilai masukan yang baru.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

- Gerbang keluaran (*output*) pertama perpaduan nilai lama dan nilai baru, yaitu:

$$O_t = \sigma\,(W_o\,.[h_{t-1}\,, x_t\,] + b_o\,)$$

- Gerbang keluaran (*output*) kedua dari satu sel, yaitu:

$$h_t = O_t * \tanh\,(C_t)$$

Didalam satu sel LSTM memiliki fungsi untuk mengkomputasi *hidden state*. Cel di LSTM dapat memutuskan apa yang akan disimpan dan di hapus di memory (*forget gate layer*). Selain dapat mengabungkan informasi saat ini dengan informasi sebelumnya, LSTM sangat efisien merekam informasi yang panjang.[1]

Bahasa pemrograman *Python* dipilih karena bahasa ini mempunyai banyak keunggulan khususnya untuk pemrograman berbasis *machine learning*. Seperti umumnya bahasa Java dan C, yang bersifat *Open Source*, interaktif, modular, dinamis, berbasis objek, dan lain-lain, bahasa *Python* memiliki banyak *library* yang dapat dengan mudah dipergunakan untuk *machine learning*, seperti : *Numpy* (untuk operasi vektor dan matrik), *Scikit-learn* (data analisisi dan statistik), *Pandas Data Frame* (pengolahan data

seperti layaknya Excell dan SQL), *Matplotlib* (visualisasi data grafik), dan Keras (API *neural network* yang bekerja di atas *Tensor Flow* atau *Theano*).

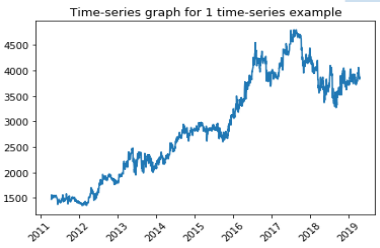
Data yang dipilih adalah data saham Telkom (TLKM), karena Telkom merupakan saham yang banyak diminati oleh umum sebagai invesatasi jangka panjang. Selain itu Telkom juga merupakan saham Badan Usaha Milik Negara (BUMN) dengan penyumbang deviden terbesar.

	date	price
0	4/18/2011	1470
1	4/19/2011	1530
2	4/20/2011	1560
3	4/21/2011	1550
4	4/25/2011	1510

	date	price
1978	4/10/2019	3930
1979	4/11/2019	3890
1980	4/12/2019	3830
1981	4/15/2019	3830
1982	4/16/2019	3870

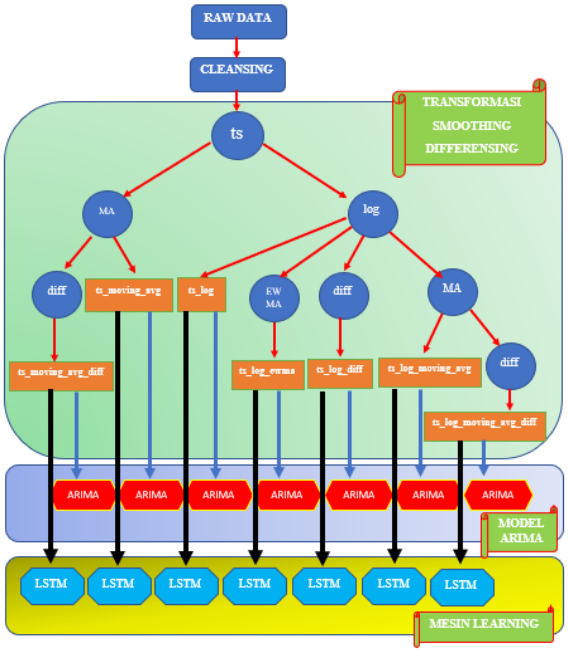
(1983, 2)

Gambar 4. Data saham Telkom



Gambar 5. Grafik perjalanan saham Telkom

Dalam penelitian ini dilakukan proses transformasi untuk mengeliminasi *trend* dan *variasi* serta mengubah data tidak stasioner menjadi stasioner. Proses transformasi ini juga berfungsi sebagai optimalisasi agar menghasilkan tingkat kesalahan sekecil mungkin. Beberapa hasil dari kombinasi transformasi masing-masing akan menjadi input untuk ARIMA dan LSTM.



Gambar 6. Bagan alur proses data

Tingkat akurasi yang tinggi diukur dengan nilai RMSE (*Root Mean Square Error*) yang rendah. Hasil olahan data dari setiap kombinasi transformasi selain bentuk numerik juga ditampilkan dalam bentuk grafik-grafik.

II. METODE PENELITIAN

A. *Pembersihan Data*

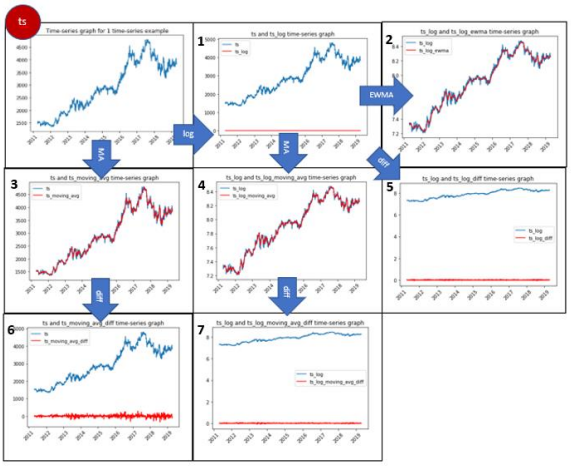
Data mentah yang diperoleh terdiri dari banyak kolom (*Date*, *Open*, *High*, *Low*, *Close*, *AdjClose*, *Volume*), untuk itu dilakukan pemilihan data yang dipergunakan dalam penelitian ini yaitu *Date* dan *Close* (*univariate*). *Date* adalah data tanggal harian dan *Close* adalah harga saham pada penutupan akhir. Pembersihan data dilakukan terhadap data-data ganda, null, pencilan (*outlier*), dan data yang tidak memiliki tipe yang sama. Penayangan data secara statistik deskriptif (percentile, kuartil, dan lainnya) juga akan membantu untuk mengetahui kecenderungan (*trend*) dan variasi data.

B. *Proses Transformasi*

Proses transformasi yang dimaksud adalah proses untuk mengeliminasi kecenderungan (*trend*), variasi data, dan penghalusan (*smoothing*). Beberapa proses transformasi yang dilakukan adalah proses merubah data menjadi data log (*log*), menggeser data rata-rata atau MA (*Moving Average*) secara mingguan, pembedaan data (*differencing*), dan menggeser data dengan menerapkan bobot secara eksponensial atau EWMA (*Exponentially Weighted Moving Averages*). Setiap proses transformasi dapat dilakukan secara

kombinatif agar data menjadi stasioner dengan tingkat akurasi yang masing-masing diukur dengan nilai RMSE. Terdapat 7 nama kombinasi transformasi yang dilakukan, sebagai berikut:

1. ts_moving_avg (data→MA)
2. ts_moving_avg_dif (data→MA→differencing)
3. ts_log (data→log)
4. ts_log_ewma (data →log→EWMA)
5. ts_log_diff (data →log→differencing)
6. ts_log_moving_avg (data →log →MA)
7. ts_log_ moving_avg _diff (data →log→MA→diff)



Gambar 7. Grafik dari 7 kombinasi transformasi

C. *Model ARIMA*

Dari setiap proses kombinasi transformasi yang telah dilakukan, ditentukan nilai parameter p, d, q untuk ARIMA terbaik dengan cara memilih nilai RMSE terkecil. Jadi dalam proses ini akan dihasilkan 7 model ARIMA yang masing-masing memiliki nilai RMSE.

D. *Mesin belajar LSTM*

Dari proses 7 kombinasi transformasi tersebut juga dipergunakan sebagai input ke LSTM. Sehingga dalam proses ini dihasilkan 7 hasil LSTM dengan masing-masing memiliki nilai RMSE.

III. HASIL PENELITIAN

A. *Data*

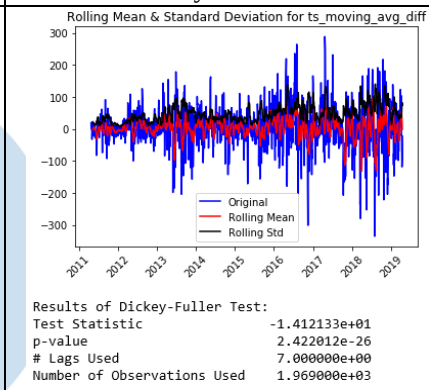
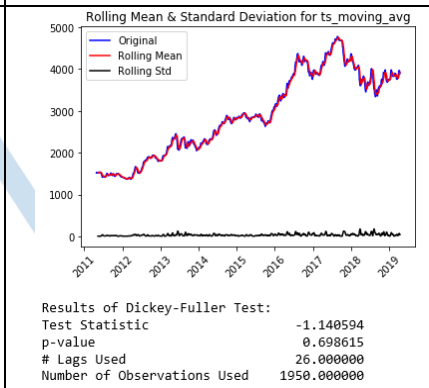
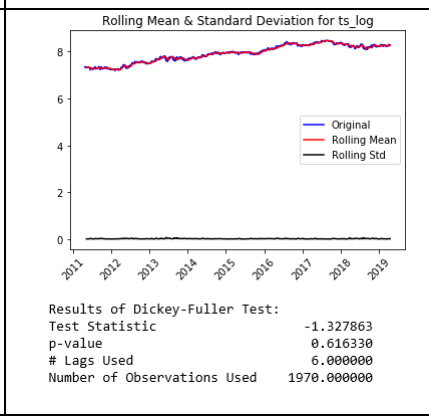
Data harian diperoleh dari https://finance.yahoo.com, setelah melalui tahap pembersihan data, jumlah data adalah 1983 mulai dari tanggal 18-4-2011 sampai dengan 16-4-2019 (Gambar 4) dan bentuk grafik saham Telkom terlihat seperti Gambar 5.

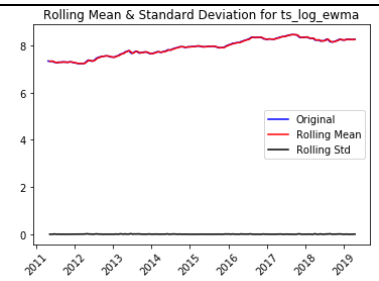
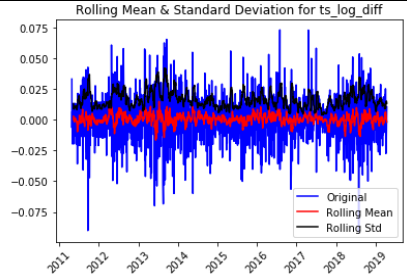
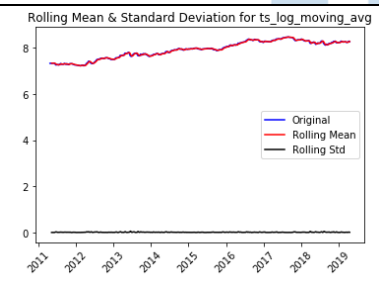
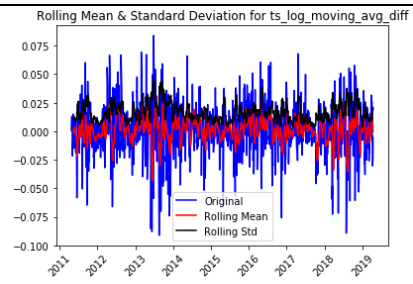
B. *Transformasi*

Hasil dari 7 kombinasi transformasi dengan nilai RMSE masing -masing dapat terlihat di Gambar 7.

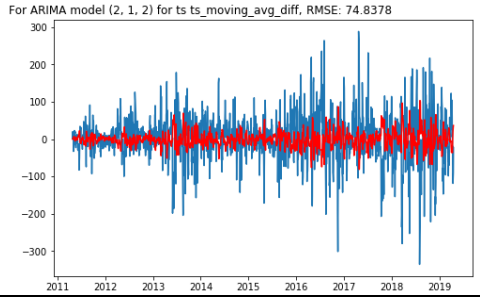
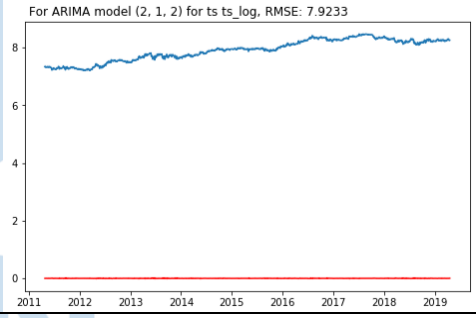
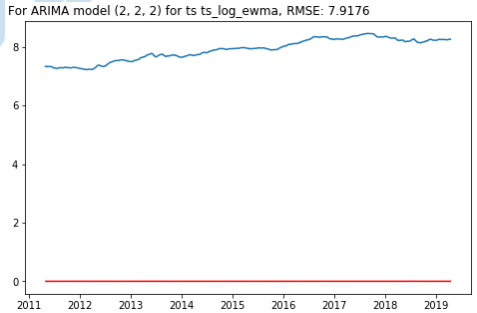
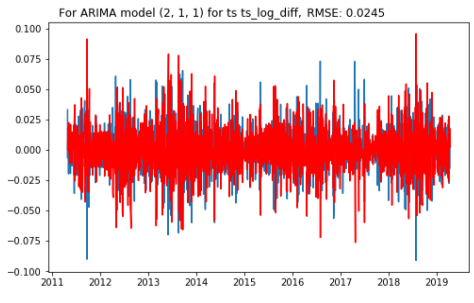
Untuk mengetahui tingkat ke stationeran data, maka dilakukan proses *rolling mean*, standar deviasi, dan penghitungan Dickey Fuller *test* ke setiap 7 kombinasi transformasi dengan hasil grafik dan RMSE pada Tabel 1.

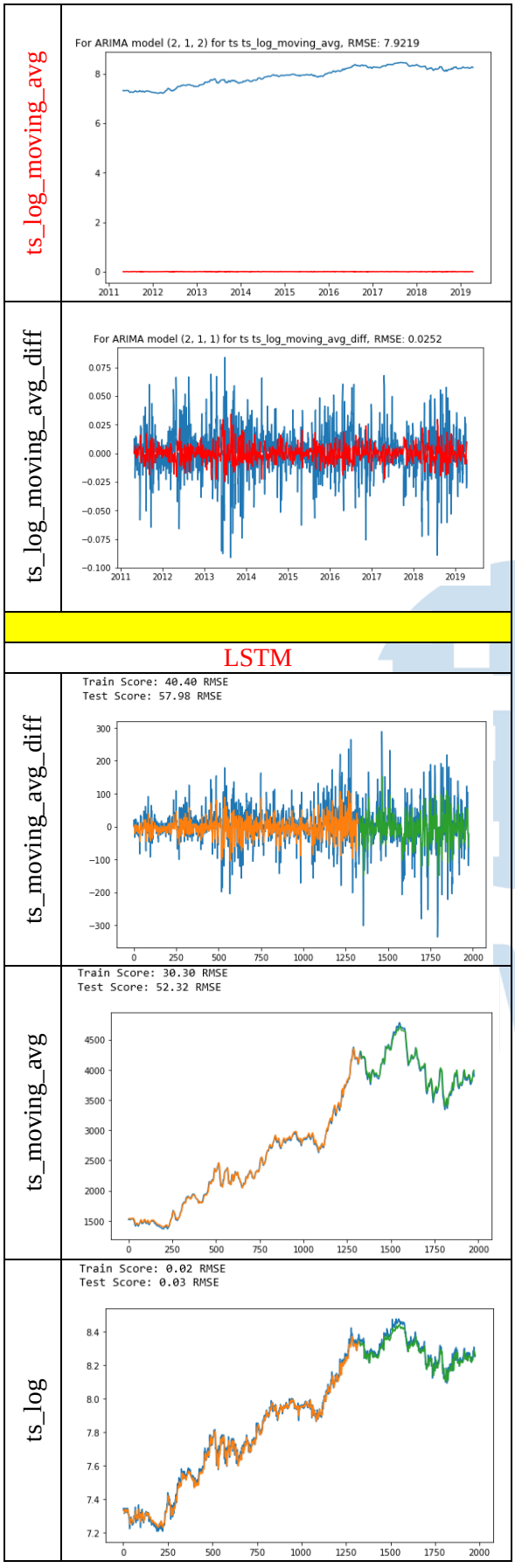
Tabel 1. Hasil *rolling*, standar deviasi dan Dickey Fuller *test*

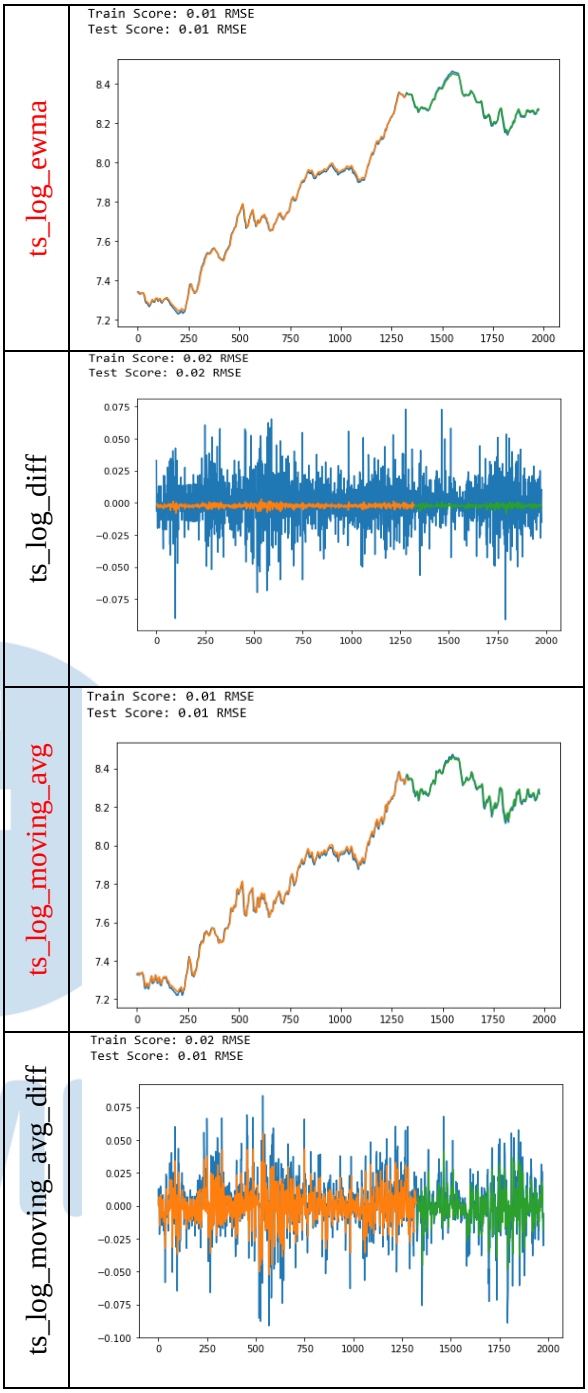
	Rolling Mean, Standar Deviasi, Dickey Fuller *Test*
ts_moving_avg_diff	
ts_moving_avg	
ts_log	

ts_log_ewma	 Results of Dickey-Fuller Test: Test Statistic -1.291568 p-value 0.633021 # Lags Used 5.000000 Number of Observations Used 1971.000000
ts_log_diff	 Results of Dickey-Fuller Test: Test Statistic -21.935568 p-value 0.000000 # Lags Used 5.000000 Number of Observations Used 1971.000000
ts_log_moving_avg	 Results of Dickey-Fuller Test: Test Statistic -1.342773 p-value 0.609384 # Lags Used 26.000000 Number of Observations Used 1950.000000
ts_log_moving_avg_diff	 Results of Dickey-Fuller Test: Test Statistic -1.395632e+01 p-value 4.604528e-26 # Lags Used 7.000000e+00 Number of Observations Used 1.969000e+03

Tabel 2. Hasil grafik dan RMSE untuk ARIMA dan LSTM

ARIMA	
ts_moving_avg_diff	
ts_moving_avg	
ts_log	
ts_log_ewma	
ts_log_diff	





C. *Model ARIMA*

Dari hasil 7 kombinasi transformasi masing-masing di cari nilai ARIMA terbaik untuk setiap kombinasi transformasi. Terlihat pada Tabel 2, kolom ARIMA dengan nilai RMSE terkecil adalah:

Tabel 3. ARIMA dan nilai RSME

Transformasi	ARIMA	RMSE
ts_log_diff	(2,1,1)	0.0245
ts_log_moving_avg_diff	(2,1,1)	0.0252

D. Mesin Belajar LSTM

Dari hasil 7 kombinasi transformasi masing-masing dipergunakan sebagai data input ke mesin belajar LSTM. Terlihat pada Tabel 2, kolom LSTM dengan nilai RMSE terkecil adalah:

Tabel 4. LSTM dan nilai RSME

Transformasi	RMSE Train	RMSE Test
ts_log_ewma	0.01	0.01
ts_log_moving_avg	0.01	0.01

IV. SIMPULAN

- Dalam penelitian ini, prediksi saham Telkom mencapai tingkat akurasi yang sangat baik yaitu 99% (RMSE=1%), setelah data melalui proses transformasi *loging* kemudian EWMA (ts_log_ewma) atau *loging* kemudian *Moving Average*.
- Hasil dari penelitian untuk prediksi data *time series* ini memperlihatkan mesin belajar LSTM lebih akurat dibandingkan dengan menggunakan model statistik ARIMA.
- Pemilihan proses transformasi yang sesuai untuk model ARIMA dan LSTM masih perlu dikembangkan agar memperolah hasil yang lebih baik lagi.
- Pengujian dengan menggunakan tipe data *time series* lainnya (selain data saham Telkom) masih perlu terus dilakukan, untuk lebih memastikan hasil dari penelitian ini.

DAFTAR PUSTAKA

[1] Olah, Christopher. *"Understanding LSTM Networks". http://colah.github.io/posts/2015-08-Understanding-LSTMs/* (2015).

[2] Hochreiter, S. & Schmidhuber, J., 1997, *Long Short-Term Memory, Neural Computation*, 8, 9, 1735–1780.

[3] Neelabh Pant, Sept 7, 2017. "A Guide For Time Series Prediction Using Recurrent Neural Networks (LSTMs)", https://blog.statsbot.co/time-series-prediction-using-recurrent-neural-networks-lstms-807fa6ca7f.

[4] Jason Brownlee, November 14, 2018 **."**How to Develop LSTM Models for Time Series Forecasting**",** https://machinelearningmastery.com/category/deep-learning-time-series.

[5] Brownlee, Jason on December 19, 2016 in Time Series. "How To Backtest Machine Learning Models for Time Series Forecasting ". https://machinelearningmastery.com/backtest-machine-learning-models-time-series-forecasting.

[6] L. J. Tashman. "Out-of-sample tests of forecasting accuracy: an analysis and review". *International Journal of Forecasting*, 16(4):437–450, 2000. https://ideas.repec.org/a/eee/intfor/v16y2000i4p437-450.html.

[7] S. Varma and R. Simon. "Bias in error estimation when using cross-validation for model selection". *BMC Bioinformatics*, 7(1):91, Feb 2006. ISSN 1471–2105. doi: 10.1186/1471–2105- 7–91.